



UNIVERSITÉ
DE MONTPELLIER



RAPPORT CONDUITE DE PROJET

Jeu Web :
Désambiguïsation lexicale



Groupe IMNA :
OUAZI Isna,
MEKAOUI Melissa,
DELALANDE Nicolas,
CULTY Alexandre

Tuteur : LAFOURCADE Mathieu

TER M1
2016 - 2017

Table des matières

1	Introduction	2
1.1	Présentation du projet	2
2	Plannings	3
2.1	Planning prévisionnel	3
2.1.1	Diagramme de <i>Gantt</i>	3
2.1.2	Précisions sur le diagramme	3
2.2	Planning réel	4
2.2.1	Diagramme de <i>Gantt</i>	4
2.3	Comparaison des plannings	4
2.4	Tâches et fonctionnalités en cours	5
3	Organisation	7
3.1	Réunions	7
3.2	Outils	7
4	Statistiques	8
4.1	Répartition des commits	8
4.1.1	Par semaine	8
4.1.2	Par heure et par jour de semaine	9
4.2	Lignes de code	9
4.3	Temps de travail	10
4.4	Résultats	10
5	Conclusion	11

1. Introduction

Ce rapport présente la partie gestion de projet du TER (Travail d'Étude Et de Recherche) réalisé dans le cadre de notre formation en première année de master informatique.

Nous précisons que les quatre membres du groupe de TER ont participé au module de conduite de projet, et que notre tuteur (M. Lafourcade) voulait que le projet soit effectué en suivant les méthodes agiles. Malheureusement, au début du semestre cette méthode ne nous était pas familière, et nous ne savions pas vraiment de quoi il s'agissait. Cependant, après l'intervention de différents acteurs du domaine dans le cadre du module de gestion de projet, nous nous sommes rendus compte que sans forcément le savoir nous avons suivi certaines des règles agiles.

1.1 Présentation du projet

Le titre du projet était : « Jeu web d'annotation de textes pour la désambiguïsation lexicale ». Ce projet avait pour but de réaliser un jeu de type GWAP (game with a purpose), jouable sur web et sur mobile, dont les données récoltées (ressources lexicales) serviraient au traitement automatique du langage naturel (TALN).

Concrètement, notre jeu permettrait de produire un corpus de phrases/textes annotés sémantiquement. Chaque terme ambigu (ex : livre) serait associé à une ou plusieurs gloses permettant d'identifier son sens correct (poids, objet, monnaie, verbe...). Le jeu collecterait l'ensemble des réponses des joueurs et distribuerait des points en fonction de l'adéquation de celles-ci.

2. Plannings

2.1 Planning prévisionnel

2.1.1 Diagramme de *Gantt*

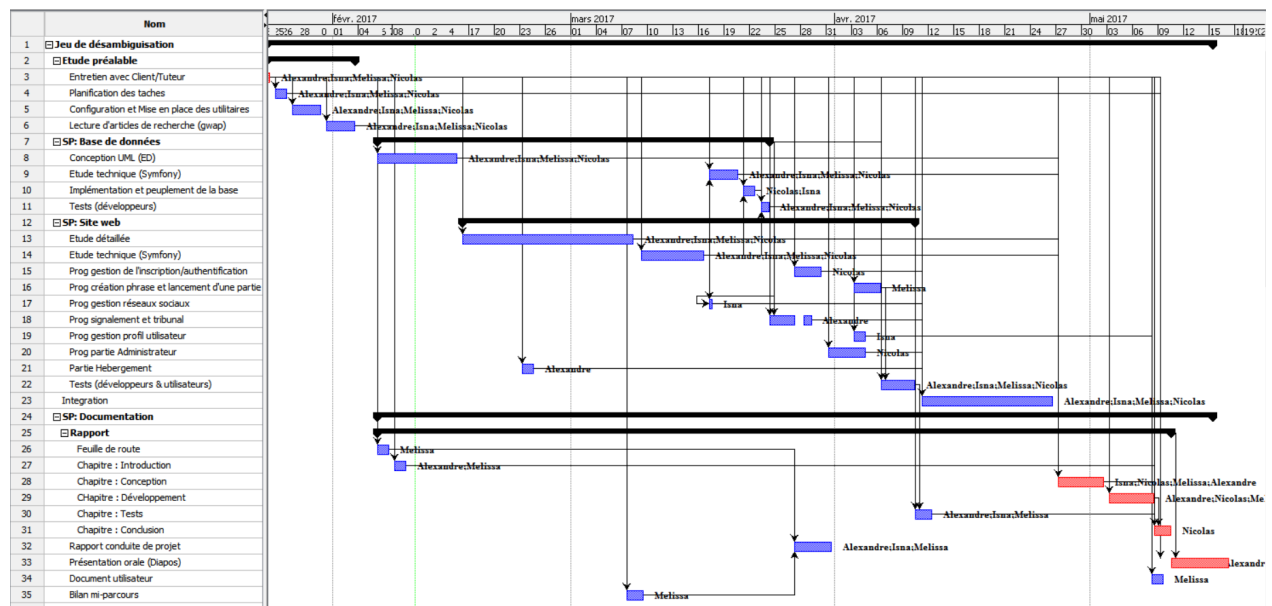


FIGURE 2.1 – Planning prévisionnel

2.1.2 Précisions sur le diagramme

Lors de la mise en place du diagramme de *Gantt* ci-dessus, nous avons décidé de découper notre projet en quatre grandes parties :

- **Étude préalable** : Comme nous l'avons appris en cours de conduite de projet, au début de chaque projet une étude préalable du domaine sur lequel nous travaillerons doit être effectuée afin de mieux comprendre le contexte et le cadre du projet. Dans notre cas, ce sont les jeux sérieux ou *gwap* (game with a purpose) qui nous intéressent. De ce fait nous avons chacun lu des livres et des articles portants sur le sujet¹. Nous avons également défini les besoins de l'utilisateur et mis en place différents diagrammes (cas d'utilisations, séquences, activités...) permettant de mieux comprendre et de mieux nous représenter le projet.
- **Base de données** : Comme dans presque toutes les applications, la base de données est un élément central de notre application. Il faut donc veiller aux choix que l'on

1. La liste des documents lus est disponible sur la feuille de route

prend vis à vis de celle-ci. Justement, nous devons utiliser une technologie qu’aucun des membres du groupe n’avaient déjà utilisé, *Symfony*. Ce framework utilise un composant nommé Doctrine qui change complètement la façon dont les données sont manipulées. C’est pour cela que nous avons prévu une étude technique dans la partie base de données.

- **Site Web** : Dans le même esprit que pour la base de données, nous avons prévu une grande partie d’étude technique pour prendre en main *Symfony* et ses composants. Nous avons ensuite estimé la durée approximative des grandes fonctionnalités en fonction de la difficulté que nous pensions rencontrer lors du développement de chacune.
- **Documentation** : Nous avons décidé d’inclure la rédaction des différents documents à fournir tout au long de ce semestre dans ce planning afin de rédiger chacun d’entre eux au fur et à mesure et de ne pas devoir le faire au dernier moment.

2.2 Planning réel

2.2.1 Diagramme de *Gantt*

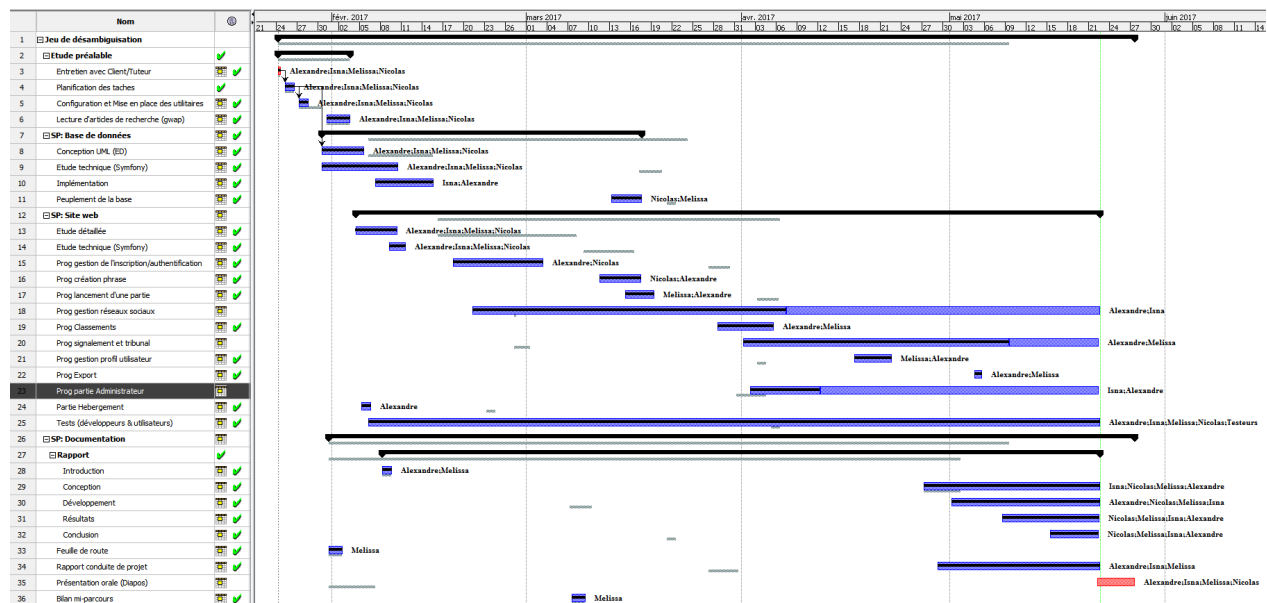


FIGURE 2.2 – Planning réel

2.3 Comparaison des plannings

Au cours du développement de l’application, nous avons décidé de modifier quelques tâches et d’en rajouter de nouvelles que nous n’avions pas prévu dans le diagramme précédent. Tout d’abord nous avons séparé la programmation de la création d’une phrase et de lancement d’une partie, ajouté la programmation de classements et de l’export et supprimé les tests d’intégration.

- **Étude préalable** : Cette phase s’est déroulée comme nous l’avions programmé, il n’y a donc rien à signaler.
- **Base de données** : Tout d’abord nous avons séparé en deux parties l’implémentation

et le peuplement de la base car elles n'ont pas été réalisées en même temps et que le peuplement de la base ne doit pas forcément intervenir juste après l'implémentation. Nous avons également supprimé les tests puisqu'il n'y avait pas lieu de les réaliser à ce moment là. Concernant le temps, on remarque que l'on avait pu commencer avant la date prévu et que l'on a aussi fini avant la date prévisionnelle.

- **Site Web** : Dans ce sous projet nous avons séparé en deux la partie création de phrase et lancement d'une partie car nous avons rencontrés de gros problèmes au niveau du lancement d'une partie avec les formulaires Symfony. Nous avons rajouté une tâche de programmation des classements très importants pour attiser l'esprit de compétition des joueurs ainsi qu'une tâche de programmation de l'export permettant à tout le monde de récupérer les données récoltées. Nous voyons malheureusement que trois tâches n'ont pas pu être achevées, la gestion des réseaux sociaux, les signalements et la partie administrateur. Pour ce qui est de la gestion des réseaux sociaux nous avons réussi à utiliser le système de partage mais pas de connexion à cause de certains problèmes techniques dû à l'utilisation du protocole OAuth. Les signalements sont fonctionnels, cependant nous n'avons pas eu le temps de réaliser la partie qui permet à tous les joueurs de voter si oui ou non un signalement est justifié ce qui doit permettre de faciliter le travail des modérateurs. Enfin, la partie administrateur était une des fonctionnalités les moins prioritaire puisque nous pouvions faire des changements directement en base si cela était nécessaire, mais nous avons quand même réalisé une page de statistiques.
- **Documentation** : Nous avons supprimé la documentation utilisateur car celle-ci se trouve directement sur le site via des bulles d'informations, remonté d'un niveau la feuille de route qui était en sous projet du rapport par erreur et renommé le chapitre des tests en résultats dans le rapport. Pour ce qui est du temps, nous étions en retard pour certains chapitres du rapport et aussi pour ce rapport de conduite de projet. En effet, nous avions prévu de réaliser progressivement les différents chapitres à partir de fin avril et de réaliser ce rapport fin mars, cependant, nous devions effectuer au même moment des travaux dans le cadre d'autres UE ce qui nous a empêché de nous concentrer pleinement sur ces tâches, nous avons donc décidé de toutes les effectuer au même moment.

Il faut aussi noter que les plannings ne prennent pas en compte les autres enseignements qui nous demandaient aussi du temps et nous ont alors empêché de nous concentrer pleinement sur les dernières fonctionnalités restantes à développer.

2.4 Tâches et fonctionnalités en cours

Le site dispose actuellement des fonctionnalités les plus importantes du jeu. Les utilisateurs peuvent jouer, créer des phrases, les signaler et même les corriger avant leur mise en ligne ou après s'ils bénéficient du statut de modérateur. Les données récoltées sont également mises à leur disposition. Toutefois certaines fonctionnalités n'ont pas été totalement achevées ou n'ont pas pu être développées à temps. Parmi celles-ci nous trouvons :

- L'inscription et la connexion via les réseaux sociaux
- Le système de vote public pour les signalements
- La gestion du site et des utilisateurs depuis la page d'administration

Quant aux idées d'amélioration, nous avons pensé à :

- Envoyer des mails automatiques afin d'attirer les joueur absents depuis un certain temps vers le site
- Créer des classements hebdomadaires dans le but de motiver les nouveaux joueurs afin qu'ils ne soient pas démotivés par la différence de score entre eux et les premiers joueurs du classement général
- Suggérer des mots ambigus oubliés lors de la création d'une phrase par un joueur
- Créer des succès, par exemple : jouer toutes les phrases ou encore créer 100 phrases, etc.
- Mettre en place un système "anti-trolls" qui bannirait les utilisateurs qui tenteraient de fausser les réponses

3. Organisation

Chaque membre du groupe travaillait principalement seul chez lui mais pouvait demander de l'aide lors de nos diverses réunions avec M. Lafourcade, ou via les différents outils que nous utilisions. Il nous ait aussi arrivé quelques fois de nous retrouver en dehors de la faculté pour travailler ensembles sur le projet.

3.1 Réunions

Chaque semaine ou presque, nous organisions une réunion avec tous les membres du groupe et M. Lafourcade. Nous avons ainsi réalisé environ dix réunions qui avaient pour but de montrer les évolutions du projet, mais aussi de demander d'éventuelles précisions ou conseils sur certains aspects du jeu.

À la fin des premières réunions, nous rédigeons un compte rendu ¹ afin d'avoir une trace écrite de celle-ci car des éléments importants y figuraient. Nous avons par la suite arrêté les comptes rendus car les réunions abordaient de plus en plus des détails de développement, nous les déposons alors directement sur l'outil des tâches à réaliser (Trello).

3.2 Outils

Pour l'organisation de notre projet, nous avons utilisé plusieurs outils afin de gérer divers aspects de celui-ci :

- Git via la plateforme GitHub ² comme logiciel de versionning pour partager le code source du projet
- LucidChart pour réaliser les diagrammes UML des cas d'utilisation et de modélisation de la base de données
- Trello pour lister les tâches à réaliser sous forme de post-it
- GoogleDoc pour récupérer les avis des utilisateurs ³ à propos du site
- Facebook messenger pour communiquer entre nous
- ShareLatex pour réaliser les différents documents et rapports

Outre le fait que ces outils répondent à des besoins, ce sont des outils gratuits et collaboratifs, nous pouvions donc tous les quatre travailler en même temps dessus sans problèmes.

1. Comptes rendus de réunion : <https://goo.gl/XIh8uR>

2. GitHub du projet : <https://github.com/alexdu98/Ambiguss>

3. Document des avis utilisateurs : <https://goo.gl/cqAM4U>

4. Statistiques

Dans cette partie nous aborderons les détails statistiques du projet, nous verrons entre autres les soumissions de code source au répertoire Git, le nombre de lignes de code, le temps que nous avons passé sur le projet...

4.1 Répartition des commits

Nous avons effectué 349 *commits* entre le 25 janvier et le 18 mai. Voici les détails de ces soumissions :

4.1.1 Par semaine

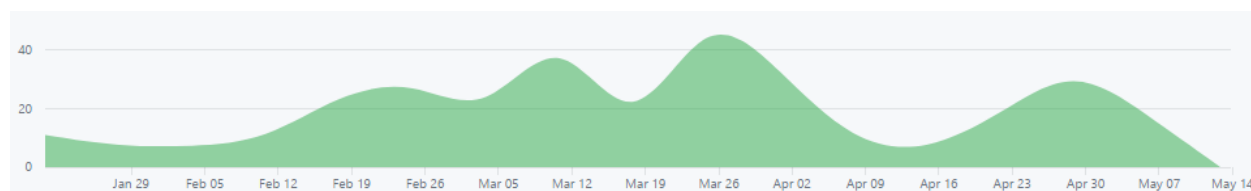


FIGURE 4.1 – Répartition des *commits* par semaine

Nous constatons sur ce graphique l'augmentation du nombre de *commits* aux alentours du 8 février, qui correspond au début des vacances d'hiver, et à la fin de la période d'analyse et d'étude technique.

Nous voyons ensuite une baisse du nombre de *commits* vers le 15 mars, soit moins de deux semaines avant les rattrapages du premier semestre qui ont eu lieu vers le 28 mars.

C'est aussi à cette date que l'on enregistre le nombre de *commits* le plus élevé puisque nous n'étions pas tous concernés par les rattrapages et que certains ont alors profité de cette semaine pour avancer sur le projet.

On remarque ensuite début avril que le nombre de *commits* chute sévèrement pendant environ quinze jours qui correspondent aux vacances de printemps. En effet, certains membres avaient un travail important à rendre à la rentrée, d'autres sont partis de Montpellier et n'avaient pas accès facilement à Internet, expliquant ainsi cette baisse.

Enfin, nous voyons le nombre de *commits* baisser en cette période de fin de projet.

4.1.2 Par heure et par jour de semaine

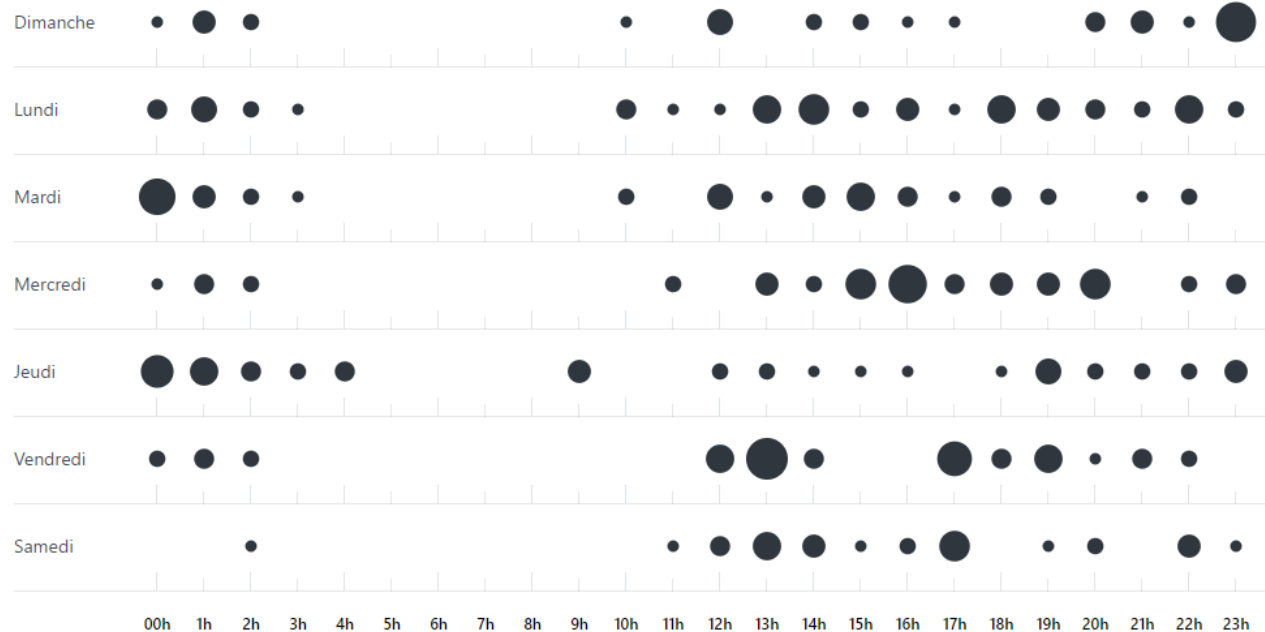


FIGURE 4.2 – Répartition des commits par heure et par jour de semaine

Nous remarquons ici que la plus part des *commits* ont eux lieux entre midi et 3h du matin, aussi bien la semaine, que le week-end. En effet, ayant le plus souvent cours le matin et en début d’après midi, nous étions plus disposé à travailler en fin d’après midi et soirée, voir même la nuit de temps en temps.

4.2 Lignes de code

Symfony dispose de plusieurs milliers de fichiers, nous ne les prenons évidemment pas en comptes ici. Une ligne de code peut aussi être un commentaire ou un saut de ligne.

Pour ce projet nous avons produit **environ 14536 lignes** de code. Détaillons un peu celles-ci :

- **322 lignes dans 6 routeurs.** Un *routeur* est un fichier de configuration *yml* permettant de lier une adresse URL à une action.
- **2560 lignes dans 13 contrôleurs.** Les contrôleurs sont les classes où se trouve le code logique.
- **5358 lignes dans 25 entités.** Une entité est une classe représentant un objet pouvant être sauvegardé en base de données. On y trouve beaucoup de commentaires car nous avons utilisé les outils de Symfony qui documente chaque méthode automatiquement.
- **1249 lignes dans 24 formulaires.** Ce sont les classes qui permettent de générer les champs d’un formulaire. Il n’y a presque pas de commentaires, mais là aussi nous avons utilisé les outils de Symfony de génération automatique.
- **703 lignes dans 9 repository.** Un repository est une classe où se trouve nos méthodes personnelles de récupérations d’un objet en base de données.
- **3441 lignes dans 50 vues.** Les vues sont les fichiers HTML qui sont affichées aux visiteurs. Une vue peut aussi contenir du code *JavaScript* propre à la page en question.
- **154 lignes dans 5 fichiers de configuration.**

- **199 lignes dans 5 services.** Un service est une classe pouvant être appelée depuis n’importe où dans le code.
- **323 lignes de CSS** pour le style du site.
- **227 lignes de JavaScript.**
- **80 lignes dans 1 commande.** Les commandes sont des classes qui peuvent être appelées depuis la console *Symfony*.
- **32 lignes dans 1 auditeur d’événement.** Un auditeur d’événement est une classe appelée à certains moments.
- **3 lignes dans 1 fichier de traduction.**

4.3 Temps de travail

Concernant le temps passé sur le projet nous estimons à (chiffres approximatifs) :

- une quinzaine d’heures (chacun) le temps d’analyse et conception.
- une quinzaine d’heures (chacun) le temps d’étude technique et de documentation nécessaire à la réalisation du projet.
- environ 480 heures (total) de développement.
- environ 9 heures (chacun) de réunion.
- une dizaine d’heures (chacun) de rédaction de rapports.

4.4 Résultats

En quelques chiffres Ambiguus c’est actuellement :

- **61 membres, 1807 visites**
- **332 phrases, 502 mots ambigus, 1278 gloses/annotations**
- **9869 réponses, 3180 parties**

5. Conclusion

Tout au long de notre projet de désambiguïsation lexicale, nous avons utilisé les outils et les pratiques vus en cours de gestion de projet tels que la planification du projet en utilisant le diagramme de *Gantt*, le partage du code source de notre travail sur un logiciel de *versionning*, l'utilisation des *post-it* via *Trello*, etc. Cependant, dès le début du projet notre tuteur M. Lafourcade a émis le souhait de travailler en Agile, il a donc fallu s'y atteler sans toutefois connaître totalement les concepts de cette méthode.

Les conférences *Wembla* bien qu'un petit peu tardives, nous ont aidé à mieux cerner les concepts des méthodes Agiles, notamment grâce aux ateliers et aux exercices de réflexion qui nous ont été proposés, ce qui nous a également permis de mieux comprendre pourquoi l'Agile était préférable à la méthode *prédictive* pour la réalisation de notre projet, la réponse étant que pour un tel projet, les changements étant très fréquents, il fallait s'y adapter facilement ou encore présenter des démonstrations le plus souvent possible au client.

Un exemple des notions que nous avons apprises grâce à ces conférences et qui nous auraient servi plus tôt dans la gestion du projet est celui du client qui émet de nouvelles idées régulièrement. En effet avec l'utilisation du GoogleDoc nous accordions beaucoup d'importance à ces retours utilisateurs qui parfois étaient moins prioritaires que d'autres fonctionnalités qui auraient dû être développées avant ces changements.

En définitive, ce projet nous aura apporté une nouvelle expérience non négligeable en termes de connaissances techniques, mais aussi en termes de gestion de projet grâce à l'application des principes théoriques vus en cours.